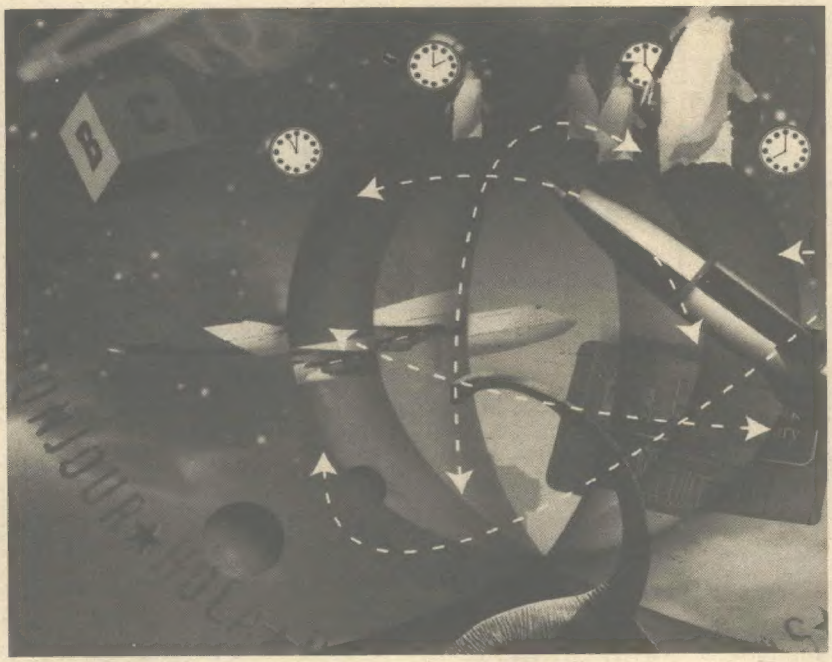
สุทธิศักดิ์ พงศ์ธนาพาณิช

เปิดโลกมัลติมีเดียด้วย VB/Win

ตอน การเล่นไฟล์ .WAV ด้วยฟังก์ชัน API ระดับล่าง (ตอนที่ 2)



การแกะรอยข้อมูลในส่วนต่าง ๆ ของไฟล์ .wav ซึ่งเป็นเทคนิคที่โปรแกรมเมอร์ใช้ในการตรวจสอบค่าพารามิเตอร์ต่าง ๆ

บทความในตอนที่ผ่านมาผู้เขียนได้เขียนถึงวิธีการตรวจสอบความสามารถและคุณสมบัติในการจัดการกับข้อมูลหรือสัญญาณเสียงของอุปกรณ์รับส่งเสียง ส่วนในบทความตอนนี้ก็เป็นตอนที่ต่อเนื่องกัน โดยในตอนนี้ผู้เขียนจะกล่าวต่อไปถึงการแกะรอยข้อมูลในส่วนต่าง ๆ (chunk) ของไฟล์ .wav ซึ่งเป็นเทคนิคที่โปรแกรมเมอร์ส่วนใหญ่ใช้ในการตรวจสอบค่าพารามิเตอร์ต่าง ๆ ที่จัดเก็บเอาไว้ในไฟล์ .wav ซึ่งไฟล์ .wav นี้จะถูกสร้างโดยดีไวซ์ไดรเวอร์นั่นเอง ดังนั้นในบทความตอนนี้ผู้อ่านก็จำเป็นต้องเรียนรู้เพิ่มเติมเกี่ยวกับฟังก์ชันในกลุ่มของ multimedia file I/O (ไม่มีการกล่าวถึงในตอนที่ 1) ซึ่งเป็นฟังก์ชันที่ขึ้นต้นด้วย mmio แต่ก่อนที่เราจะไปถึงจุดนั้นผู้เขียนก็ขออธิบายคร่าว ๆ เกี่ยวกับโครงสร้างของไฟล์ RIFF เสียก่อน เพื่อให้ผู้อ่านมีความเข้าใจถึงการจัดเก็บข้อมูลสัญญาณเสียงในไฟล์ .wav ดียิ่งขึ้น ซึ่งมีรายละเอียดต่อไปนี้

โครงสร้างของไฟล์ RIFF

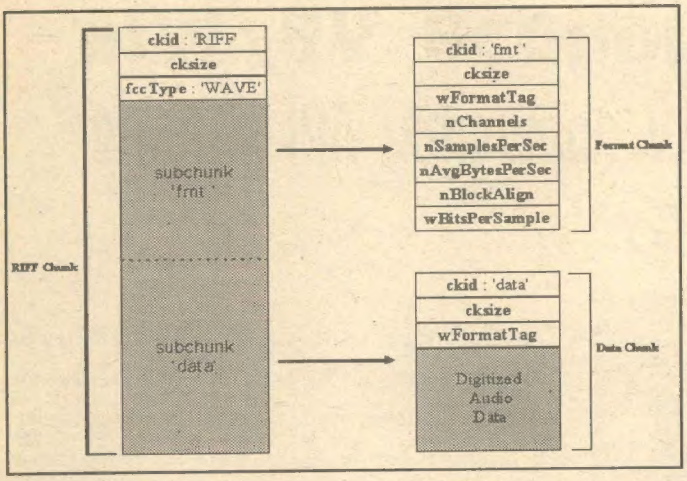
ไฟล์ RIFF (RIFF ย่อมาจากคำว่า Resource Interchange File Format) เป็นไฟล์ที่ได้รับการพัฒนาสำหรับการจัดเก็บข้อมูลด้านมัลติมีเดียโดยเฉพาะ โดยที่ข้อมูลที่จัดเก็บลงในรูปแบบของไฟล์ RIFF จะจัดเก็บเป็นบล็อก ๆ โดยแต่ละบล็อกเราเรียกว่า Chunk สำหรับไฟล์ RIFF ที่มากับวินโดวส์ 3.1 จะมีโครงสร้างทั่วไปดังรูปที่ 1

ผู้อ่านจะเห็นว่าในไฟล์ .wav หนึ่ง ๆ จะประกอบด้วยบล็อกของข้อมูลที่ชื่อ RIFF ซึ่งเป็นบล็อกที่ใหญ่ที่สุด โดยจะประกอบด้วยฟิลด์ข้อมูล 5 ฟิลด์ ดังนี้ ckid, cksize, fccType, subchunk 'fmt ' และ subchunk 'data' และในการอ่านข้อมูลของ 'RIFF' ไม่ว่าจะเป็นบล็อกใดก็ตาม จะกระทำโดยการใช้ข้อมูลโครงสร้าง MMCKINFO ดังมีรายละเอียดต่อไปนี้

ข้อมูลโครงสร้าง MMCKINFO

เป็นข้อมูลโครงสร้างสำหรับจัดเก็บข้อมูลของบล็อกข้อมูล RIFF โดยที่แต่ละฟิลด์มีรายละเอียดดังนี้

```
Type MMCKINFO
     ckid          As FOURCC     ' Chunk ID
     cksize        As Long       ' Chunk size
     fccType       As FOURCC     ' Form type or list type
     dwDataOffset  As Long       ' Offset of data portion of chunk
     dwFlags       As Long       ' Flags used by MMIO functions
End Type
```



รูปที่ 1 แสดงโครงสร้างทั่วไปของไฟล์ RIFF

ckid

ข้อมูลชนิดสตริงขนาด 4 ไบต์ ซึ่งฟิลด์นี้ได้ถูกประกาศให้เป็นข้อมูลชนิด FOURCC ซึ่งข้อมูลโครงสร้างนี้ก็จะประกอบด้วยสมาชิกเพียงฟิลด์เดียว ดังนี้

```
Type FOURCC
    FourBytes       As String * 4
End Type
```

cksize

ข้อมูลชนิดเลขจำนวนเต็ม long บอกขนาดของฟิลด์ข้อมูลของ chunk โดยที่ค่าของ cksize จะเท่ากับขนาดของฟิลด์ข้อมูล และไม่รวม 4 ไบต์สำหรับฟิลด์ ckid และ 4 ไบต์สำหรับฟิลด์ cksize

fccType

ข้อมูลชนิดสตริงขนาด 4 ไบต์ เช่นเดียวกับฟิลด์ ckid ซึ่งจะจัดเก็บสตริงที่บอกถึงประเภทของ form สำหรับบล็อก RIFF และบอกถึงประเภทของ list สำหรับบล็อก LIST

dwDataOffset

ข้อมูลชนิดเลขจำนวนเต็ม long บอกระยะห่าง (จำนวนไบต์) จากจุดเริ่มต้นของไฟล์ถึงจุดเริ่มต้นของฟิลด์ข้อมูลของบล็อก RIFF

dwFlags

ข้อมูลชนิดเลขจำนวนเต็ม long ซึ่งเราจะใช้กำหนดเงื่อนไขการทำงานของฟังก์ชันไฟล์มัลติมีเดีย I/O ซึ่งผู้อ่านสามารถกำหนดค่าได้ตามค่าคงที่ดังตารางที่ 1

ตารางที่ 1

ค่าคงที่	ค่าตัวเลข	รายละเอียด
MMIO_READ	&H0	กำหนดให้เปิดไฟล์สำหรับการอ่านเท่านั้น
MMIO_WRITE	&H1	กำหนดให้เปิดไฟล์สำหรับการเขียนเท่านั้น
MMIO_READWRITE	&H2	กำหนดให้เปิดไฟล์สำหรับทั้งการเขียนและการอ่าน
MMIO_COMPAT	&H0	กำหนดให้เปิดไฟล์ในโหมดของการคอมแพตทิเบิล
MMIO_EXCLUSIVE	&H10	กำหนดให้เปิดไฟล์ในโหมด exclusive เพื่อใช้งานส่วนตัว
MMIO_DENYWRITE	&H20	ให้เปิดไฟล์ โดยไม่ให้มีการเขียนทับจากแอพพลิเคชันอื่น
MMIO_DENYREAD	&H30	กำหนดให้เปิดไฟล์ โดยไม่ให้มีการอ่านจากแอพพลิเคชันอื่น
MMIO_DENYNONE	&H40	กำหนดให้เปิดไฟล์ โดยสามารถอ่านหรือเขียนทับได้จากแอพพลิเคชันอื่น

แต่สำหรับในส่วนถัดไปของ RIFF ที่เรียกว่า subchunk 'fmt ' หรือ format chunk นั้น จะมีความแตกต่างจากบล็อกอื่น ๆ อยู่บ้าง ดังนั้นเราจะอ่านข้อมูลในบล็อกนี้ได้โดยใช้ข้อมูลโครงสร้าง PCMWAVE FORMAT ดังมีรายละเอียดดังต่อไปนี้

ข้อมูลโครงสร้าง PCMWAVEFORMAT

โดยที่ข้อมูลต่าง ๆ ที่จัดเก็บในบล็อกนี้จะเป็นข้อมูลที่จะถูกใช้ในการคำนวณและบอกคุณลักษณะของข้อมูลของสัญญาณเสียงในรูปแบบ PCM สำหรับข้อมูลโครงสร้าง PCMWAVEFORMAT จะประกอบด้วยสมาชิกเพียง 2 ฟิลด์ ดังนี้

```
Type PCMWAVEFORMAT
    wf              As WAVEFORMAT
    wBitsPerSample  As Integer
End Type
```

wf

เป็นข้อมูลชนิดข้อมูลโครงสร้าง WAVEFORMAT ซึ่งแต่ละสมาชิกมีรายละเอียดดังนี้

```
Type WAVEFORMAT
    wFormatTag          As Integer  ' Format type
    nChannels           As Integer  ' Number of channels (i.e. mono, stereo, etc.)
    nSamplesPerSec      As Long     ' Sample rate
    nAvgBytesPerSec     As Long     ' For buffer estimation
    nBlockAlign         As Integer  ' Block size of data
End Type
```

ข้อมูลชนิดเลขจำนวนเต็ม กำหนดชนิดของรูปแบบ ซึ่งในปัจจุบันได้มีการกำหนดเพียงรูปแบบเดียว ดังนี้

ค่าคงที่	ค่าตัวเลข	รายละเอียด
WAVE_FORMAT_PCM	1	ข้อมูลในรูปแบบ PCM

nChannels

ข้อมูลชนิดเลขจำนวนเต็ม กำหนดจำนวนแชลแนลของข้อมูล โดยที่ข้อมูลโมโนจะใช้เพียง 1 แชลแนล ในขณะที่ข้อมูลสเตอริโอจะใช้ 2 แชลแนล

nSamplesPerSec

ข้อมูลชนิดเลขจำนวนเต็ม long กำหนดอัตราการสุ่มข้อมูล ในหน่วยของอัตราการสุ่มข้อมูลใน 1 วินาที

nAvgBytesPerSec

ข้อมูลชนิดเลขจำนวนเต็ม long กำหนดขนาดของข้อมูลเฉลี่ยในหน่วยของไบต์ สำหรับการขนถ่ายในช่วงเวลา 1 วินาที

nBlockAlign

ข้อมูลชนิดเลขจำนวนเต็ม กำหนดขอบเขตการจัดเรียง (alignment) บล็อกในหน่วยของไบต์ โดยที่ข้อมูลในรูปแบบ PCM จะเรียงข้อมูลตามจำนวนไบต์ที่ใช้ในการสุ่มแต่ละครั้ง เช่น ถ้าหากข้อมูลเป็นแบบสเตอริโอ nBlockAlign ก็จะมีค่าเท่ากับ 4 ไบต์ เป็นต้น (ให้ผู้อ่านดูค่าตัวเลขที่เกิดขึ้นได้จากการทดลองโปรแกรมตัวอย่างในตอนท้ายของบทความ)

wBitsPerSample

ข้อมูลชนิดเลขจำนวนเต็ม กำหนดจำนวนบิตที่ใช้ในการสุ่มข้อมูล 1 ครั้ง เช่น 8 หมายถึงใช้ 8 บิตในการสุ่มข้อมูลแต่ละครั้งเป็นต้น

ส่วนในบล็อก subchunk 'data' หรือที่เรียกว่า data chunk นั้นก็เป็นบล็อกที่จัดเก็บข้อมูลของสัญญาณเสียงในรูปแบบข้อมูลดิจิตอล ดังนั้นในการเล่นกลับไฟล์ .wav เราก็ต้องทำการอ่านข้อมูลทั้งหมดในบล็อกนี้ (ด้วยฟังก์ชัน mmioRead) เพื่อส่งให้กับอุปกรณ์ส่งสัญญาณเสียงต่อไป (ผู้เขียนจะอธิบายถึงวิธีการเล่นกลับไฟล์ .wav ในบทความตอนต่อไป)

ฟังก์ชันไฟล์มัลติมีเดีย I/O

ฟังก์ชันไฟล์มัลติมีเดีย I/O เป็นกลุ่มของฟังก์ชันที่สนับสนุนไฟล์ในรูปแบบ RIFF ซึ่งในปัจจุบันไมโครซอฟต์ได้จัดแบ่งออกเป็น 4 กลุ่มด้วยกัน ดังนี้

กลุ่มจัดการพื้นฐานไฟล์ด้าน I/O

ฟังก์ชันในกลุ่มนี้จะทำหน้าที่ด้านพื้นฐานของไฟล์มัลติมีเดีย เช่น การเปิด-ปิดไฟล์ การอ่าน-เขียนไฟล์ เป็นต้น ประกอบด้วย 6 ฟังก์ชันดังนี้

mmioClose	ทำหน้าที่ปิดไฟล์
mmioOpen	ทำหน้าที่เปิดไฟล์
mmioRead	ทำหน้าที่อ่านข้อมูลจากไฟล์
mmioRename	ทำหน้าที่เปลี่ยนชื่อไฟล์
mmioSeek	ทำหน้าที่ย้ายตำแหน่งปัจจุบันในไฟล์
mmioWrite	ทำหน้าที่เขียนข้อมูลลงในไฟล์

กลุ่มจัดการด้านไฟล์บัฟเฟอร์ I/O

ฟังก์ชันในกลุ่มนี้จะทำหน้าที่ด้านการจัดการบัฟเฟอร์ ซึ่งเป็นพื้นที่สำรองสำหรับการอ่านหรือเขียนข้อมูลระหว่างไฟล์กับอุปกรณ์รับ-ส่งเสียง ซึ่งประกอบด้วย 5 ฟังก์ชันดังนี้

mmioAdvance	ทำหน้าที่เติมหรือล้างข้อมูลออกจากบัฟเฟอร์สำหรับการเข้าถึงโดยตรง
mmioFlush	ทำหน้าที่เขียนข้อมูลทั้งหมดในบัฟเฟอร์ลงในไฟล์
mmioGetInfo	ทำหน้าที่อ่านข้อมูลเกี่ยวกับบัฟเฟอร์ของไฟล์
mmioSetBuffer	ทำหน้าที่เปลี่ยนขนาดของบัฟเฟอร์ของไฟล์
mmioSetInfo	ทำหน้าที่เปลี่ยนข้อมูลเกี่ยวกับบัฟเฟอร์ของไฟล์

กลุ่มทำงานร่วมกับไฟล์ RIFF

ฟังก์ชันในกลุ่มนี้จะทำหน้าที่เพิ่มเติมจากฟังก์ชันในกลุ่มจัดการพื้นฐานไฟล์ด้าน I/O โดยผู้อ่านสามารถใช้ฟังก์ชันในกลุ่มนี้ในการเลื่อนตำแหน่งไปยังส่วนต่าง ๆ (chunk) ของไฟล์ ประกอบด้วย 5 ฟังก์ชันดังนี้

mmioAscend	ทำหน้าที่ย้ายตำแหน่งออกจากบล็อกของ chunk ปัจจุบัน
mmioDescend	ทำหน้าที่ย้ายตำแหน่งเข้าไปในบล็อกของ chunk ตามที่กำหนด
mmioCreateChunk	ทำหน้าที่สร้าง chunk ในไฟล์
mmioFOURCC	ทำหน้าที่แปลงตัวอักษร 4 ตัวให้เป็นโค้ด FOURCC (สำหรับภาษาซี)
mmioStringToFOURCC	ทำหน้าที่แปลงสตริง (ASCIIZ) ให้เป็นโค้ด FOURCC (สำหรับภาษาซี)

สำหรับในบทความตอนนี้ผู้เขียนจะขออธิบายเฉพาะฟังก์ชันในกลุ่มจัดการพื้นฐานไฟล์ด้าน I/O และกลุ่มทำงานร่วมกับไฟล์ RIFF บาง

ฟังก์ชันที่เกี่ยวข้องกับโปรแกรมตัวอย่างเท่านั้น เนื่องจากโปรแกรมตัวอย่างของบทความตอนนี้จะเน้นเฉพาะเทคนิคการอ่านข้อมูลต่าง ๆ จากไฟล์ .wav ในรูปแบบ RIFF เท่านั้น ส่วนฟังก์ชันที่เหลือถ้าหากผู้อ่านท่านใดสนใจก็ขอให้เขียนจดหมายมายังตู้ป.ณ. แล้วผู้เขียนจะได้รวบรวมเอาไว้ในตอนต่อ ๆ ไป

mmioAscend

Declaration
```
Declare Function mmioAscend Lib "MMSYSTEM" (ByVal hmmio As Integer,_
    lpCk As Any, ByVal wFlags As Integer) As Integer
```

Description ทำหน้าที่ย้ายตำแหน่งออกจากบล็อกของ chunk ปัจจุบัน เช่น ในโปรแกรมตัวอย่างผู้อ่านได้ใช้ฟังก์ชันนี้ในการย้ายตำแหน่งปัจจุบันออกจากบล็อก 'fmt '

Parameters
- `hmmio` ตัวแปรชนิดเลขจำนวนเต็ม กำหนดหมายเลข handle ของไฟล์ที่ถูกเปิด
- `lpCk` ตัวแปรชนิดข้อมูลโครงสร้าง MMCKINFO
- `wFlags` ตัวแปรชนิดเลขจำนวนเต็ม โดยทั่วไปจะกำหนดเป็น 0

Example
```
ErrCode% = mmioAscend(hMMIO%, MMCkInfoChild, 0)
```

Comment ถ้าหากฟังก์ชันนี้ส่งกลับค่าเท่ากับ 0 แสดงว่าฟังก์ชันทำงานสำเร็จ หรือมิฉะนั้นก็จะคืนกลับเป็นโค้ดของข้อผิดพลาดซึ่งสามารถเป็นไฟได้ 2 โค้ดดังนี้

ค่าคงที่	ค่าตัวเลข	คำอธิบาย
MMIOERR_CANNOTWRITE	(256 + 6)	ไม่สามารถเขียนข้อมูลในบัฟเฟอร์ลงในไฟล์
MMIOERR_CANNOTSEEK	(256 + 7)	มีการย้ายตำแหน่งไปยังจุดสิ้นสุดของ chunk

mmioDescend

Declaration
```
Declare Function mmioDescend Lib "MMSYSTEM" (ByVal hmmio As Integer,_
    lpCk As Any,lpCkParent As Any, ByVal wFlags As Integer) As Integer
```

Description ทำหน้าที่ย้ายตำแหน่งเข้าไปในบล็อกของ chunk ตามที่กำหนด

Parameters
- `hmmio` ตัวแปรชนิดเลขจำนวนเต็ม กำหนดหมายเลข handle ของไฟล์ที่ถูกเปิด
- `lpck` ตัวแปรชนิดข้อมูลโครงสร้าง MMCKINFO โดยถ้าหาก wFlags มีค่าเท่ากับ MMIO_FINDRIFF พารามิเตอร์นี้ก็จะหมายถึง chunk แม่ทันที และผู้อ่านต้องกำหนดให้พารามิเตอร์ lpCkParent เท่ากับ NULL
- `lpCkParent` ตัวแปรชนิดข้อมูลโครงสร้าง MMCKINFO โดยถ้าหาก wFlags มีค่าเท่ากับ MMIO_FINDCHUNK พารามิเตอร์นี้ก็จะหมายถึง chunk แม่ทันที และพารามิเตอร์ lpCk ก็จะเท่ากับ chunk ภายในทันที
- `wFlags` ตัวแปรชนิดเลขจำนวนเต็ม กำหนดเงื่อนไขในการค้นหา chunk ซึ่งผู้อ่านสามารถกำหนดได้ดังต่อไปนี้

ค่าคงที่	ค่าตัวเลข	คำอธิบาย
MMIO_FINDCHUNK	&H10	ค้นหา chunnk ตามค่า ID ที่ได้กำหนด ให้กับสมาชิก ckid ของข้อมูลโครงสร้าง MMCKINFO
MMIO_FINDRIFF	&H20	ค้นหา chunnk ที่มี ID เท่ากับ RIFF
MMIO_FINDLIST	&H40	ค้นหา chunk ที่มี ID เท่ากับ LIST

Example
```
Dim MMCKInfoParent As MMCKINFO
Dim MMCkInfoChild As MMCKINFO
Dim hMMIO%, ErrCode%
```

→

```
hMMIO% = mmioOpen(filename$, ByVal 0&, MMIO_READ)
If hMMIO% <> 0  Then
     ' Find WAVE Parent Chunk
      MMCKInfoParent.fccType.FourBytes = "WAVE"
      ErrCode% = mmioDescend(hMMIO%, MMCKInfoParent, ByVal 0&, MMIO_FINDRIFF)
     If ErrCode% = 0 Then
          ' Find fmt Chunk
      MMCkInfoChild.ckid.FourBytes = "fmt "
      ErrCode% = mmioDescend(hMMIO%, MMCkInfoChild, MMCKInfoParent, _
                                  MMIO_FINDCHUNK)
      End IF
Endif
```

Comment ค่าที่ส่งกลับโดยฟังก์ชันนี้จะบอกถึงความสำเร็จของฟังก์ชัน ดังนี้

0 หมายถึง ฟังก์ชันทำงานสำเร็จ

<>0 หมายถึง ฟังก์ชันทำงานล้มเหลว

mmioClose

Declaration
```
Declare Function mmioClose Lib "MMSYSTEM" (ByVal hmmio As Integer, _
      ByVal wFlags As Integer) As Integer
```

Description ทำหน้าที่ปิดไฟล์

Parameters
- hmmio ตัวแปรชนิดเลขจำนวนเต็ม กำหนดหมายเลข handle ของไฟล์ที่ถูกเปิด
- wFlags ตัวแปรชนิดเลขจำนวนเต็ม โดยทั่วไปจะกำหนดเป็น 0

Example
```
ErrCode% = mmioClose(hMMIO%, 0)
```

Comment ถ้าหากฟังก์ชันทำงานสำเร็จก็จะคืนกลับค่าเท่ากับ 0 แต่ถ้าหากทำงานไม่สำเร็จก็จะคืนกลับรหัสข้อผิดพลาด MMIOERR_CANNOTWRITE ซึ่งโดยปกติแสดงว่าในบัฟเฟอร์ยังคงมีข้อมูลที่ต้องการเขียนลงในไฟล์เหลืออยู่นั่นเอง

mmioOpen

Declaration
```
Declare Function mmioOpen Lib "MMSYSTEM" (ByVal szFilename As String,_
            lpMMIOINFO As Any, ByVal dwOpenFlags As Long) As Integer
```

Description ทำหน้าที่เปิดไฟล์มัลติมีเดียตามชื่อที่กำหนด

Parameters
- szFilename ตัวแปรชนิดสตริง กำหนดชื่อไฟล์ (สามารถรวมไดเรคทรอรีได้ด้วย) มัลติมีเดียที่ต้องการเปิด
- lpMMIOINFO ตัวแปรชนิดข้อมูลโครงสร้าง MMIOINFO แต่โดยทั่วไปเราจะกำหนดให้พารามิเตอร์นี้มีค่าเท่ากับ NULL
- dwOpenFlags ตัวแปรชนิดเลขจำนวนเต็ม long กำหนดเงื่อนไขในการเปิดไฟล์ ซึ่งผู้อ่านสามารถกำหนดให้มีค่าได้ตามที่ได้กล่าวมาแล้วข้างต้นในค่าของฟิลด์ dwFlags ของข้อมูลโครงสร้าง MMCKINFO

Example
```
Dim hMMIO%
hMMIO% = mmioOpen(filename$, ByVal 0&, MMIO_READ)
```

Comment ค่าที่ส่งกลับโดยฟังก์ชันนี้จะบอกถึงความสำเร็จของฟังก์ชัน ดังนี้

0 หมายถึง ฟังก์ชันทำงานไม่สำเร็จ

>0 หมายถึง หมายเลข handle สำหรับไฟล์มัลติมีเดียที่ถูกเปิด ซึ่งค่าที่ได้จากฟังก์ชันนี้ผู้อ่านจะไม่สามารถนำไปใช้กับคำสั่งอื่น ๆ ของ VB/Win นอกเหนือจากฟังก์ชันในกลุ่มไฟล์มัลติมีเดีย I/O เช่น Open, Close หรือ Get เป็นต้น

mmioRead

Declaration
```
Declare Function mmioRead Lib "MMSYSTEM" (ByVal hmmio As Integer,_
        pch As Any, ByVal cch As Long) As Long
```

Description ทำหน้าที่อ่านข้อมูลตามจำนวนไบต์ที่กำหนด จากไฟล์

Parameters
- hmmio — ตัวแปรชนิดเลขจำนวนเต็ม กำหนดหมายเลข handle ของไฟล์ที่ถูกเปิด
- pch — ตัวแปรชนิดใด ๆ ก็ได้ สำหรับถูกกำหนดเป็นบัฟเฟอร์สำหรับการจัดเก็บข้อมูลที่อ่านได้จากไฟล์
- cch — ตัวแปรชนิดเลขจำนวนเต็ม long กำหนดจำนวนไบต์ที่ต้องการอ่านจากไฟล์

Example
```
Dim MMCKInfoParent As MMCKINFO
Dim MMCkInfoChild As MMCKINFO
Dim PCMWaveFmtRecord As PCMWAVEFORMAT
Dim hMMIO%, ErrCode%, ReadByte&

      hMMIO% = mmioOpen(filename$, ByVal 0&, MMIO_READ)
      If hMMIO% <> 0 Then
   ' Find WAVE Parent Chunk
   MMCKInfoParent.fccType.FourBytes = "WAVE"
   ErrCode% = mmioDescend(hMMIO%, MMCKInfoParent, ByVal 0&, MMIO_FINDRIFF)
   If ErrCode% = 0 Then
      ' Find fmt Chunk
      MMCkInfoChild.ckid.FourBytes = "fmt "
      ErrCode% = mmioDescend(hMMIO%, MMCkInfoChild, MMCKInfoParent,_
                 MMIO_FINDCHUNK)
      If ErrCode% = 0 Then
 ' Read PCM Wave Format Record
 ReadByte& = mmioRead(hMMIO%, PCMWaveFmtRecord, MMCkInfoChild.cksize)
End IF
End If
End If
```

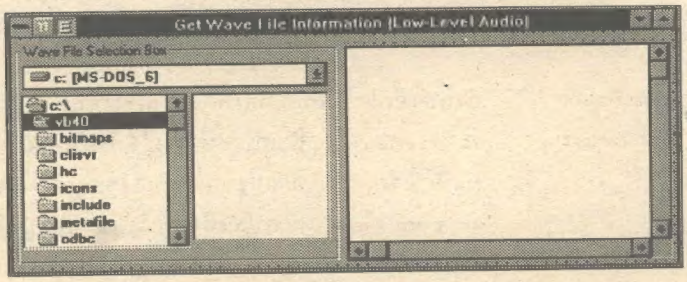
Comment ค่าที่ส่งกลับโดยฟังก์ชันนี้จะบอกถึงความสำเร็จของฟังก์ชัน ดังนี้
- -1 หมายถึง ฟังก์ชันทำงานไม่สำเร็จ
- 0 หมายถึง ไม่มีข้อมูลที่สามารถอ่านจากไฟล์
- >0 หมายถึง จำนวนไบต์ของข้อมูลที่ถูกอ่านขึ้นมาจากไฟล์ ซึ่งโดยปกติจะต้องมีค่าเท่ากับค่าของพารามิเตอร์ cch ที่กำหนดให้กับฟังก์ชัน

โปรแกรม WAVEINFO

โปรแกรมตัวอย่าง WAVEINFO เป็นโปรแกรมที่ใช้แสดงข้อมูลต่าง ๆ ที่เก็บไว้ในไฟล์ .wav ซึ่งเป็นไฟล์ที่เป็นไปตามรูปแบบมาตรฐาน RIFF โปรแกรมนี้ประกอบด้วยฟอร์ม Form1 เพียงฟอร์มเดียว โดยในขณะออกแบบจะมีหน้าตาดังในรูปที่ 2 โดยที่ผู้เขียนจะใช้ Text1 ทางขวามือของฟอร์มในการแสดงผลข้อมูลต่าง ๆ ที่อ่านได้จากไฟล์ .wav ที่ถูกเลือกจาก File1 ในกลุ่มของ Wave File Selection Box

และเมื่อผู้อ่านทดลองรันโปรแกรม ผู้อ่านสามารถเลือกที่จะอ่านข้อมูลจากไฟล์ .wav ใด ๆ ก็ได้ โดยให้ผู้อ่านเลือกไฟล์ .wav ที่ต้องการ จากนั้นก็จะปรากฏข้อมูลของแต่ละ chunk ของไฟล์ที่ถูกเลือกใน Text1 ทันที โดยในรูปที่ 3 แสดงถึงข้อมูลที่อ่านได้จากไฟล์ tada.wav

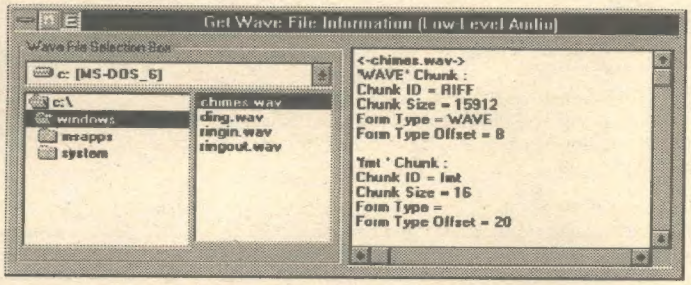
ส่วนการทำงานของโปรแกรมก็จะเริ่มต้นที่เมื่อผู้อ่านรันตัวอย่าง รูทีน Form_Load ก็จะถูกเรียกทันที ซึ่งภายในรูทีนนี้ผู้อ่านก็ได้เขียนโค้ดสำหรับการย้ายฟอร์มมาแสดงผลยังกึ่งกลางของหน้าจอดังนี้



รูปที่ 2 แสดงฟอร์ม Form1 ในขณะออกแบบ

```
Move (Screen.Width - Scalewidth) \2,_
(Screen.Height - Scaleheight) \ 2
```

จากนั้นโปรแกรมก็จะรอให้ผู้อ่านเลือกไฟล์ .wav ที่ต้องการ โดยเมื่อมีการเลือกไฟล์ ก็จะทำให้เกิดเหตุการณ์ File1_Click ทันที ซึ่งภาย



รูปที่ 3 แสดงตัวอย่างผลลัพธ์ของโปรเจ็กต์ WAVEINFO.VBP ในขณะทำงาน

ในรูทีนก็จะเป็นการอ่านชื่อไฟล์ที่ถูกเลือกพร้อมทั้งไดเรคทอรี จากนั้นก็เรียกรูทีน GetWaveFileInfo ซึ่งเป็นรูทีนหัวใจที่สำคัญของโปรแกรม เพื่อทำการอ่านข้อมูลของ chunk ต่าง ๆ ของไฟล์ที่ถูกเลือก ดังนี้

```
filename$ = File1.List(File1.ListIndex)
GetWaveFileInfo filename$
```

โดยการทำงานของรูทีน GetWaveFileInfo ก็จะเริ่มต้นด้วยการเปิดไฟล์ .wav ที่ต้องการด้วยฟังก์ชัน mmioOpen จากนั้นก็ทำการตรวจสอบผลลัพธ์ที่ได้ โดยถ้าหากค่าของ hMMIO% เท่ากับ 0 แสดงว่าเกิดข้อผิดพลาดในการเปิดไฟล์ ก็จะกระโดดออกจากรูทีนทันที

```
Dim MMCKInfoParent As MMCKINFO
Dim MMCkInfoChild As MMCKINFO
Dim PCMWaveFmtRecord As PCMWAVEFORMAT
Dim hMMIO%, ErrCode%, ReadByte&, p$

p$ = "<-" & filename$ & "->" & vbCrLf
hMMIO% = mmioOpen(filename$, ByVal 0&, MMIO_READ)
If hMMIO% = 0 Then
    p$ = p$ & "-ERROR-" & vbCrLf & vbCrLf
    ErrCode% = True
    MsgBox "ไม่สามารถเปิดไฟล์ .wav", 16
Else
```

แต่ถ้าหาก hMMIO% มีค่ามากกว่า 0 เราก็จะเริ่มต้นขบวนการอ่านข้อมูลจาก chunk ต่าง ๆ ของไฟล์ .wav ได้เลย โดยเริ่มต้นที่การอ่านข้อมูลจาก chunk ที่มี ID ดังนี้ 'WAVE' โดยการเลื่อนตำแหน่งปัจจุบันเข้าไปยัง chunk 'WAVE' ด้วยฟังก์ชัน mmioDescend และค่าตรวจสอบผลลัพธ์ที่ได้ โดยถ้าหากค่าของ ErrCode% ไม่เท่ากับ 0 ก็แสดงว่าเกิดข้อผิดพลาด แต่ถ้าหากมีค่าเท่ากับ 0 เราก็จะทำการอ่านข้อมูลที่ได้จากตัวแปร MMCKInfoParent ทันที (เนื่องจาก chunk ที่มี FCC Type เท่ากับ 'WAVE' เป็น chunk แม่ ดังนั้นพารามิเตอร์ลำดับที่ 3 จึงมีค่าเท่ากับ 0&

```
        ' Find WAVE Parent Chunk
        p$ = p$ & "'WAVE' Chunk :" & vbCrLf
        MMCKInfoParent.fccType.FourBytes =_
"WAVE"
        ErrCode% = mmioDescend(hMMIO%,_
MMCKInfoParent, ByVal 0&, MMIO_FINDRIFF)
        If ErrCode% Then
            p$ = p$ & "-ERROR-" & vbCrLf &_
vbCrLf
            MsgBox "ค้นหา 'WAVE' chunk ไม่พบ",16
        Else
            p$ = p$ & "Chunk ID = " &_
sZtrim(MMCKInfoParent.ckid.FourBytes) & vbCrLf
            p$ = p$ & "Chunk Size = " &_
MMCKInfoParent.cksize & vbCrLf
            p$ = p$ & "Form Type = " &_
sZtrim(MMCKInfoParent.fccType.FourBytes)&_
vbCrLf
            p$ = p$ & "Form Type Offset = " &_
MMCKInfoParent.dwDataOffset & vbCrLf & vbCrLf
```

จากนั้นก็เป็นการอ่านข้อมูลจาก chunk 'fmt' ต่อไป โดยเริ่มต้นด้วยการเลื่อนตำแหน่งปัจจุบันเข้ามาใน chunk นี้ด้วยฟังก์ชัน mmioDescend เช่นเดิม แต่เนื่องจาก chunk 'fmt' เป็นบล็อกที่เก็บข้อมูลที่บอกถึงคุณสมบัติของข้อมูลสัญญาณเสียงในส่วนของ chunk 'data' ซึ่งเป็นข้อมูลแบบสุ่มในรูปแบบ PCM (Pulse Code Modulation) ซึ่งไมโครซอฟต์ได้เตรียมข้อมูลโครงสร้าง PCMWAVEFORMAT สำหรับการจัดเก็บข้อมูลเหล่านี้อยู่แล้ว ดังนั้นเราจึงอ่านข้อมูลให้กับข้อมูลโครงสร้าง PCMWAVEFORMAT ด้วยฟังก์ชัน mmioRead ดังนี้

```
        ' Find fmt Chunk
        p$ = p$ & "'fmt ' Chunk :" & vbCrLf
        MMCkInfoChild.ckid.FourBytes = "fmt "
        ErrCode% = mmioDescend(hMMIO%, MMCkInfoChild, MMCKInfoParent, MMIO_FINDCHUNK)
        If ErrCode% Then
            p$ = p$ & "-ERROR-" & vbCrLf & vbCrLf
            MsgBox "ค้นหา 'fmt' chunk ไม่พบ", 16
        Else
            p$ = p$ & "Chunk ID = " & sZtrim(MMCkInfoChild.ckid.FourBytes) & vbCrLf
            p$ = p$ & "Chunk Size = " & MMCkInfoChild.cksize & vbCrLf
            p$ = p$ & "Form Type = " & sZtrim(MMCkInfoChild.fccType.FourBytes) & vbCrLf
            p$ = p$ & "Form Type Offset = " & MMCkInfoChild.dwDataOffset & vbCrLf & vbCrLf


            ' Read PCM Wave Format Record
            p$ = p$ & "PCM Wave Format :" & vbCrLf
            ReadByte& = mmioRead(hMMIO%, PCMWaveFmtRecord, MMCkInfoChild.cksize)
            If ReadByte& <= 0 Then
                ErrCode% = True
```

```
                p$ = p$ & "-ERROR-" & vbCrLf & vbCrLf
                MsgBox "ไม่สามารถอ่านรูปแบบของ 'WAVE'", 16
            Else
                p$ = p$ & "Format Type = " & PCMWaveFmtRecord.wf.wFormatTag & vbCrLf
                p$ = p$ & "Number of Channels = " & PCMWaveFmtRecord.wf.nChannels & vbCrLf
                p$ = p$ & "Sample Rate = " & PCMWaveFmtRecord.wf.nSamplesPerSec & vbCrLf
                p$ = p$ & "Average Byte Per Second = " & _
PCMWaveFmtRecord.wf.nAvgBytesPerSec & vbCrLf
                p$ = p$ & "Data Block Size = " & PCMWaveFmtRecord.wf.nBlockAlign & vbCrLf
                p$ = p$ & "Bits Per Sample = " & PCMWaveFmtRecord.wBitsPerSample & _
vbCrLf & vbCrLf
```

ต่อมาก็เป็นการอ่านข้อมูลจากส่วนของ chunk 'data' แต่ก่อนอื่นผู้อ่านต้องทำการเลื่อนตำแหน่งปัจจุบันออกจาก chunk 'fmt' เสียก่อนด้วยฟังก์ชัน mmioAscend จากนั้นจึงค่อยอ่านข้อมูลด้วยฟังก์ชัน mmioDescend อีกครั้ง และสุดท้ายก็เป็นการปิดไฟล์ .wav ที่ถูกเปิดด้วยฟังก์ชัน mmioClose

```
                ' Ascend back one level in the RIFF file.
                p$ = p$ & "'data' Chunk :" & vbCrLf
                ErrCode% = mmioAscend(hMMIO%, MMCkInfoChild, 0)
                If ErrCode% Then
                    p$ = p$ & "-ERROR-" & vbCrLf & vbCrLf
                    MsgBox "ไม่สามารถเข้าถึง 'fmt' chunk", 16
                Else
                    ' Read data chunk.
                    MMCkInfoChild.ckid.FourBytes = "data"
                    ErrCode% = mmioDescend(hMMIO%, MMCkInfoChild, MMCKInfoParent, _
MMIO_FINDCHUNK)
                    If ErrCode% Then
                        p$ = p$ & "-ERROR-" & vbCrLf & vbCrLf
                        MsgBox "¤é¹ËÒ 'data' chunk ไม่พบ", 16
                    Else
                        p$ = p$ & "Chunk ID = " & sZtrim(MMCkInfoChild.ckid.FourBytes) _
                          & vbCrLf
                        p$ = p$ & "Chunk Size = " & MMCkInfoChild.cksize & vbCrLf
                        p$ = p$ & "Form Type = " & _
            sZtrim(MMCkInfoChild.fccType.FourBytes) & vbCrLf
                        p$ = p$ & "Form Type Offset = " & _
                                  MMCkInfoChild.dwDataOffset & vbCrLf & vbCrLf
                    End If
                End If
            End If
        End If
        ErrCode% = mmioClose(hMMIO%, 0)
        If ErrCode% Then MsgBox "ไม่สามารถปิดไฟล์ .wav", 16
    End If
    Text1.Text = p$
```

ลิสติงไฟล์โปรเจ็กต์ WAVEINFO.VBP

```
Form=waveinfo.frm
ProjWinSize=122,0,248,127
ProjWinShow=2
IconForm="Form1"
Title="Waveinfo"
Name="Waveinfo"
HelpContextID="0"
StartMode=0
VersionCompatible32="0"
MajorVer=1
MinorVer=0
RevisionVer=0
AutoIncrementVer=0
ServerSupportFiles=0
VersionComments="Multimedia Programming with VB/Win"
VersionCompanyName="Microcomputer Magazine SE-ED"
VersionFileDescription="Get WAVE File Information By Low-Level Audio API Function."
VersionLegalCopyright="Sutthisak Phongthanapanich"
```

ส่วนในบรรทัดสุดท้ายของรูทีนนี้ ก็เป็นการนำข้อมูลของตัวแปร p$ มาแสดงผลลงใน Text1 ซึ่งผู้อ่านจะเห็นว่าด้วยวิธีนี้คุณก็จะสามารถศึกษาถึงโครงสร้างของไฟล์ .wav ในรูปแบบมาตรฐาน RIFF ได้ไม่ยากเลยครับ

ส่วนรายละเอียดที่เหลือผู้อ่านก็สามารถศึกษาได้จากโปรแกรมตัวอย่างของบทความนี้ และฟังก์ชันที่เหลือผู้เขียนก็จะขอนำไปอธิบายในตอนต่อไป พบกันใหม่ในฉบับหน้าครับ

ลิสติง ไฟล์ฟอร์ม
WAVEINFO.FRM

```
VERSION 4.00
Begin VB.Form Form1
   BorderStyle     =   3  'Fixed Dialog
   Caption         =   "Get Wave File Information (Low-Level Audio)"
   ClientHeight    =   3165
   ClientLeft      =   825
   ClientTop       =   1650
   ClientWidth     =   8085
   BeginProperty Font
      name            =   "MS Sans Serif"
      charset         =   222
      weight          =   700
      size            =   8.25
      underline       =   0   'False
      italic          =   0   'False
      strikethrough   =   0   'False
   EndProperty
   ForeColor       =   &H80000008&
   Height          =   3570
   Left            =   765
   LinkTopic       =   "Form1"
   MaxButton       =   0   'False
   MinButton       =   0   'False
   ScaleHeight     =   3165
   ScaleWidth      =   8085
   Top             =   1305
   Width           =   8205
   Begin VB.TextBox Text1
      Height          =   2895
      Left            =   4095
      Locked          =   -1  'True
      MultiLine       =   -1  'True
      ScrollBars      =   3  'Both
      TabIndex        =   4
      Top             =   180
      Width           =   3915
   End
   Begin VB.Frame Frame2
      Caption         =   " Wave File Selection Box "
      BeginProperty Font
         name            =   "MS Sans Serif"
         charset         =   222
         weight          =   400
         size            =   8.25
         underline       =   0   'False
         italic          =   0   'False
         strikethrough   =   0   'False
      EndProperty
      Height          =   2985
      Left            =   60
      TabIndex        =   0
      Top             =   75
      Width           =   3930
      Begin VB.DirListBox Dir1
         Height          =   2055
         Left            =   75
         TabIndex        =   3
         Top             =   720
         Width           =   2025
      End
      Begin VB.DriveListBox Drive1
         Height          =   330
         Left            =   90
         TabIndex        =   2
         Top             =   315
         Width           =   3720
      End
```

ลิสติงไฟล์ฟอร์ม WAVEINFO.FRM (ต่อ)

```
      Begin VB.FileListBox File1
         Height          =   2010
         Left            =   2160
         Pattern         =   "*.wav"
         TabIndex        =   1
         Top             =   720
         Width           =   1650
      End
   End
End
Attribute VB_Name = "Form1"
Attribute VB_Creatable = False
Attribute VB_Exposed = False
Option Explicit
'
' Project Name  : WAVEINFO.VBP
' Program Name  : Get WAVE File Information By Low-Level Audio API Function.
' File name     : WAVEINFO.FRM
' Written by    : SUTTHISAK PHONGTHANAPANICH
'
Private Const WAVE_MAPPER = -1           ' Device ID for Wave Mapper.
Private Const WAVE_FORMAT_PCM = 1        ' Flags for wFormatTag field of WAVEFORMAT.

' Read/Write mode numbers. (bit field MMIO_RWMODE)
Private Const MMIO_READ = &H0                    ' Open file for reading only
Private Const MMIO_WRITE = &H1                   ' Open file for writing only
Private Const MMIO_READWRITE = &H2               ' Open file for reading and writing

' various MMIO flags.
Private Const MMIO_FINDCHUNK = &H10              ' mmioDescend: find a chunk by ID
Private Const MMIO_FINDRIFF = &H20               ' mmioDescend: find a LIST chunk
Private Const MMIO_FINDLIST = &H40               ' mmioDescend: find a RIFF chunk

' General waveform format structure. (information common to all formats)
Private Type WAVEFORMAT
    wFormatTag      As Integer      ' Format type
    nChannels       As Integer      ' Number of channels (i.e. mono, stereo, etc.)
    nSamplesPerSec  As Long         ' Sample rate
    nAvgBytesPerSec As Long         ' For buffer estimation
    nBlockAlign     As Integer      ' Block size of data
End Type

' Specific waveform format structure for PCM data.
Private Type PCMWAVEFORMAT
    wf              As WAVEFORMAT
    wBitsPerSample  As Integer
End Type

Private Type FOURCC
    FourBytes       As String * 4
End Type

' RIFF chunk information data structure.
Private Type MMCKINFO
    ckid            As FOURCC    ' Chunk ID
    cksize          As Long      ' Chunk size
    fccType         As FOURCC    ' Form type or list type
    dwDataOffset    As Long      ' Offset of data portion of chunk
    dwFlags         As Long      ' Flags used by MMIO functions
End Type

Private Declare Function mmioAscend Lib "MMSYSTEM" (ByVal hMMIO As Integer, _
        lpCk As Any, ByVal wFlags As Integer) As Integer
Private Declare Function mmioClose Lib "MMSYSTEM" (ByVal hMMIO As Integer, _
        ByVal wFlags As Integer) As Integer
Private Declare Function mmioDescend Lib "MMSYSTEM" (ByVal hMMIO As Integer, _
        lpCk As Any, lpCkParent As Any, ByVal wFlags As Integer) As Integer
Private Declare Function mmioOpen Lib "MMSYSTEM" (ByVal szFilename As String, _
        lpMMIOINFO As Any, ByVal dwOpenFlags As Long) As Integer
Private Declare Function mmioRead Lib "MMSYSTEM" (ByVal hMMIO As Integer, _
        pch As Any, ByVal cch As Long) As Long

Private Sub Dir1_Change()
    File1.Path = Dir1.Path
```

→

อิสติง ไฟล์ฟอร์ม WAVEINFO.FRM (ต่อ)

```
    ChDir Dir1.Path
End Sub

Private Sub Drive1_Change()
    Dir1.Path = Drive1.Drive
    ChDrive (Drive1.Drive)
End Sub

Private Sub File1_Click()
    Dim filename$

    filename$ = File1.List(File1.ListIndex)
    GetWaveFileInfo filename$
End Sub

Private Sub Form_Load()
    ' Show form in the middle of screen.
    Move (Screen.Width - ScaleWidth) \ 2, (Screen.Height - ScaleHeight) \ 2
End Sub

Private Sub Form_Unload(Cancel As Integer)
    End
End Sub

Private Sub GetWaveFileInfo(filename$)
    Dim MMCKInfoParent As MMCKINFO
    Dim MMCkInfoChild As MMCKINFO
    Dim PCMWaveFmtRecord As PCMWAVEFORMAT
    Dim hMMIO%, ErrCode%, ReadByte&, p$

    p$ = "<-" & filename$ & "->" & vbCrLf
    hMMIO% = mmioOpen(filename$, ByVal 0&, MMIO_READ)
    If hMMIO% = 0 Then
        p$ = p$ & "-ERROR-" & vbCrLf & vbCrLf
        ErrCode% = True
        MsgBox "ไม่สามารถเปิดไฟล์ .wav", 16
    Else
        ' Find WAVE Parent Chunk
        p$ = p$ & "'WAVE' Chunk :" & vbCrLf
        MMCKInfoParent.fccType.FourBytes = "WAVE"
        ErrCode% = mmioDescend(hMMIO%, MMCKInfoParent, ByVal 0&, MMIO_FINDRIFF)
        If ErrCode% Then
            p$ = p$ & "-ERROR-" & vbCrLf & vbCrLf
            MsgBox "ค้นหา 'WAVE' chunk ไม่พบ", 16
        Else
            p$ = p$ & "Chunk ID = " & sZtrim(MMCKInfoParent.ckid.FourBytes) & vbCrLf
            p$ = p$ & "Chunk Size = " & MMCKInfoParent.cksize & vbCrLf
            p$ = p$ & "Form Type = " & sZtrim(MMCKInfoParent.fccType.FourBytes) & vbCrLf
            p$ = p$ & "Form Type Offset = " & MMCKInfoParent.dwDataOffset & vbCrLf & vbCrLf

            ' Find fmt Chunk
            p$ = p$ & "'fmt ' Chunk :" & vbCrLf
            MMCkInfoChild.ckid.FourBytes = "fmt "
            ErrCode% = mmioDescend(hMMIO%, MMCkInfoChild, MMCKInfoParent, MMIO_FINDCHUNK)
            If ErrCode% Then
                p$ = p$ & "-ERROR-" & vbCrLf & vbCrLf
                MsgBox "ค้นหา 'fmt' chunk ไม่พบ", 16
            Else
                p$ = p$ & "Chunk ID = " & sZtrim(MMCkInfoChild.ckid.FourBytes) & vbCrLf
                p$ = p$ & "Chunk Size = " & MMCkInfoChild.cksize & vbCrLf
                p$ = p$ & "Form Type = " & sZtrim(MMCkInfoChild.fccType.FourBytes) & vbCrLf
                p$ = p$ & "Form Type Offset = " & MMCkInfoChild.dwDataOffset & vbCrLf & vbCrLf

                ' Read PCM Wave Format Record
                p$ = p$ & "PCM Wave Format :" & vbCrLf
                ReadByte& = mmioRead(hMMIO%, PCMWaveFmtRecord, MMCkInfoChild.cksize)
                If ReadByte& <= 0 Then
                    ErrCode% = True
                    p$ = p$ & "-ERROR-" & vbCrLf & vbCrLf
                    MsgBox "ไม่สามารถอ่านรูปแบบของ 'WAVE'", 16
                Else
                    p$ = p$ & "Format Type = " & PCMWaveFmtRecord.wf.wFormatTag & vbCrLf
                    p$ = p$ & "Number of Channels = " & PCMWaveFmtRecord.wf.nChannels & vbCrLf
                    p$ = p$ & "Sample Rate = " & PCMWaveFmtRecord.wf.nSamplesPerSec & vbCrLf
```

ลิสติง ไฟล์ฟอร์ม WAVEINFO.FRM (ต่อ)

```
                    p$ = p$ & "Average Byte Per Second = " & _
 PCMWaveFmtRecord.wf.nAvgBytesPerSec & vbCrLf
                    p$ = p$ & "Data Block Size = " & PCMWaveFmtRecord.wf.nBlockAlign & vbCrLf
                    p$ = p$ & "Bits Per Sample = " & PCMWaveFmtRecord.wBitsPerSample & _
 vbCrLf & vbCrLf

                    ' Ascend back one level in the RIFF file.
                    p$ = p$ & "'data' Chunk :" & vbCrLf
                    ErrCode% = mmioAscend(hMMIO%, MMCkInfoChild, 0)
                    If ErrCode% Then
                        p$ = p$ & "-ERROR-" & vbCrLf & vbCrLf
                        MsgBox "ไม่สามารถเข้าถึง 'fmt' chunk", 16
                    Else
                        ' Read data chunk.
                        MMCkInfoChild.ckid.FourBytes = "data"
                        ErrCode% = mmioDescend(hMMIO%, MMCkInfoChild, MMCKInfoParent, _
  MMIO_FINDCHUNK)
                        If ErrCode% Then
                            p$ = p$ & "-ERROR-" & vbCrLf & vbCrLf
                            MsgBox "ค้นหา 'data' chunk ไม่พบ", 16
                        Else
                            p$ = p$ & "Chunk ID = " & sZtrim(MMCkInfoChild.ckid.FourBytes) _
                              & vbCrLf
                            p$ = p$ & "Chunk Size = " & MMCkInfoChild.cksize & vbCrLf
                            p$ = p$ & "Form Type = " & _
                 sZtrim(MMCkInfoChild.fccType.FourBytes) & vbCrLf
                            p$ = p$ & "Form Type Offset = " & _
                                      MMCkInfoChild.dwDataOffset & vbCrLf & vbCrLf
                        End If
                    End If
                End If
            End If
        End If
        ErrCode% = mmioClose(hMMIO%, 0)
        If ErrCode% Then MsgBox "ไม่สามารถปิดไฟล์ .wav", 16
    End If
    Text1.Text = p$
End Sub

Private Function sZtrim(msg$) As String
    Dim i%, p$

    On Error Resume Next

    p$ = msg$
    i% = InStr(p$, Chr$(0))
    If i% Then p$ = Left$(p$, i% - 1)
    sZtrim = RTrim$(p$)
End Function
```

เทคนิคประจำคอลัมน์

ถาม ในกรณีที่พารามิเตอร์ของฟังก์ชัน API เป็นข้อมูลชนิดโครงสร้าง และเราต้องการส่งตัวแปรที่เป็นพอยน์เตอร์ หรือ NULL ให้กับพารามิเตอร์นั้น ๆ จะทำอย่างไร ?

ตอบ ก่อนอื่นเราต้องกำหนดให้พารามิเตอร์นั้นเป็นข้อมูลชนิด Any เสียก่อน เพื่อบังคับให้ VB/Win ไม่ทำการตรวจสอบชนิดของตัวแปร จากนั้นเราก็สามารถส่งค่าพอยน์เตอร์ให้กับฟังก์ชันได้โดยการนำหน้าค่าตัวเลขพอยน์เตอร์ด้วยคำสงวน ByVal ดังเช่นตัวอย่าง

```
ErrCode% = mmioDescend(hMMIO%, MMCKInfoParent, _
ByVal 0&, MMIO_FINDRIFF)
```

ในกรณีของฟังก์ชัน mmioDescend ข้างต้นพารามิเตอร์ที่ 2 และ 3 เป็นข้อมูลชนิดเดียวกัน แต่เราจะส่งข้อมูลโครงสร้าง MMCKINFO ให้กับพารามิเตอร์ที่ 2 ส่วนพารามิเตอร์ที่ 3 เราจะกำหนดให้กับพอยน์เตอร์ NULL